

Create your own YouTube

In the first of our new series of in-depth tutorials for more experienced PC users, Ben Pitt explains how to encode a video to host on your own website

There's no doubt about it, YouTube is one of the best sites on the internet. For viewers, it's packed with millions of videos on every conceivable subject within the bounds of good taste. For filmmakers, it provides an incredibly easy way to share productions with broadband-equipped computer users around the world.

However, uploading videos to YouTube is so easy and popular that doing so can feel like throwing a penny into the ocean. Furthermore, for video projects you really care about, YouTube's fixed resolution, compression ratio and webpage design can look a little unprofessional. It's possible to embed a YouTube video into your own website, but then you have to put up with the YouTube logo emblazoned on the video itself.

By hosting videos on your own website, you will have precise control over the encoding of your video to maximise quality. You will also be able to do away with YouTube's 10-minute time limit, its fixed page design and the YouTube logo. In this tutorial, we'll assume a basic grasp of HTML and we'll explain how to create and embed high-quality streaming video into an existing website.

WHICH VIDEO FORMAT?

The three leading video formats to choose from are Windows Media, Apple's QuickTime and Adobe (formerly Macromedia) Flash. All three offer excellent quality while keeping file sizes small enough to stream, but each requires you to have a compatible video player program on your computer. Needless to say, they use different ones, but Flash has the widest installation base. According to the Adobe website, Flash is installed on 98 per cent of internet-enabled computers, while Windows Media is on 83.7 per cent of computers and QuickTime 67.6 per cent of computers.

Flash Video can also be used as part of a more elaborate Flash Movie (SWF) file. This means video and vector graphics can be combined to create complex, high-resolution animations, or to add elements of interactivity (such as transport controls) to a video.

The downside of using Flash Video (FLV) files is that, by themselves, they're not compatible with web browsers. You can't drop an FLV file straight into a webpage. Instead, it must be embedded in a Flash Movie (SWF) file, which is then incorporated in a webpage. This poses a problem for people without a copy of the expensive Flash Pro program, which is used to create the SWF file. However, there are several third-party SWF files available from kind-hearted Flash programmers, designed specifically to incorporate FLVs into a website. These can be customised to your requirements without the need for Flash Pro.

BALANCING ENCODING SETTINGS

The FLV format is what is known as a wrapper format. The .flv extension indicates that it's a Flash Video file but doesn't tell us which video or audio codec is used. The two common video codecs for FLV are Sorenson

Spark (also known as H.263) and On2 VP6. Sorenson Spark is compatible with Flash Player 7 or later, and quality is roughly equivalent to Microsoft's WMV 9. On2 VP6 requires Flash Player 8 or later, but its quality is a significant step up from Sorenson Spark.

It's up to you whether you want to limit compatibility for increased quality, but we believe it's worth doing so. According to Adobe's website, Flash 7 or later is installed on 98 per cent of internet-enabled computers in Europe, while the figure is 95.9 per cent for Flash 8. The audio codec is allocated separately, and MP3 is standard in FLV files.

YouTube's videos use the Sorenson Spark codec at a fixed 320x240 resolution at up to 30 frames per second (fps), with an average bandwidth of around 250Kbit/s. Meanwhile, its audio is mono at 22.05kHz, encoded to MP3 at around 64Kbit/s. These are well-chosen settings, but they aren't perfect for every project.

So why would you want to deviate from YouTube's settings? Say that you had shot your video in widescreen. YouTube letterboxes widescreen videos, but it's much better to use a resolution with a native widescreen 16:9 ratio. Alternatively, it's possible to stretch widescreen footage so it fills a standard 4:3 screen during encoding, and then stretch it back again on playback. As such, it's not a serious problem if your chosen encoder doesn't explicitly support widescreen, as long as the encoded video doesn't have black bars.

FEEL THE BANDWIDTH

Next you must decide what bandwidth you want to use. A sensible choice is 300Kbit/s because it falls comfortably within the 512Kbit/s bandwidth of basic broadband packages. If you're confident that your intended audience has a faster connection – perhaps over an office intranet or to a small group of friends or clients – you may consider 400 or even 800Kbit/s. This will dramatically improve the quality, but think carefully about whether or not it's worth the risk of annoying viewers with unreliable playback.

Once you've decided how much bandwidth you can afford, you need to decide how you want to spend it. If your video contains audio, it will need to be allocated a share of bandwidth. CD-quality audio runs at 44.1kHz stereo, and encoding a CD-to-MP3 format at 128Kbit/s is considered by most people to be equivalent to CD quality. However, this is a sizeable chunk of your bandwidth, so it makes sense to reduce these settings.

Dropping to 64Kbit/s or lower will introduce significant compression artefacts – a tinkling noise that we can best describe as like digital wind chimes. However, dropping the sample rate to 22.05kHz mono will also give the MP3 encoder less data to compress, thereby reducing artefacts without sacrificing too much quality.

The remaining bandwidth goes to the video part of the file, but there's a complex balancing act to be made concerning smoothness (the frame rate), detail (the resolution) and compression artefacts. If you want to

INTRODUCTION

Want to put your videos online but don't care for YouTube's presentation style or technical limitations? You can do it yourself, embedding movie files on your own site, and making them as long and hi-res as you want.

In this month's brand-new *Advanced Projects*, we explore Flash videos and how you can create your own personal version of YouTube.

Ben Pitt

Music and photography expert

labs@computershopper.co.uk



have lots of high-resolution frames every second, the codec will struggle to describe each one correctly with the given bandwidth. As a result, artefacts appear, usually in the form of blocks and smudges. Bear in mind that video can be magnified on playback. Therefore, dropping the resolution will result in a loss of detail but doesn't mean your viewers will need to squint at a tiny video playback window.

The balancing act gets more complicated because modern video codecs use temporal compression. This means that, rather than describing each frame from scratch, only areas that have changed from the previous frame are described. As a result, it's much easier for codecs to compress slow-moving footage than fast-moving footage. Consequently, slow-moving sections of a video can look detailed and clean, while fast-moving sections of the same video look messy.

SIMPLE STEPS

These are a lot of criteria to consider at the same time, so let's simplify it. We'll assume we have a bandwidth of 300Kbit/s, of which 64Kbit/s will be used for audio. We will also assume a frame rate of 25fps, as this is the standard for UK video cameras. If you have shot your video with a digital camera at 30fps, use that instead.

The remaining 236Kbit/s bandwidth is enough for reasonable quality video at 320x240 pixels. However, if there is a lot of fast motion, you're better off dropping the resolution to 240x180 to reduce artefacts. Conversely, if your video is fairly inactive it's worth trying 400x300. To get your head around other scenarios, try the Video Bandwidth Estimator at <http://sorenson->

usa.com/vbe, which gives recommended bandwidths depending on frame rate, resolution and amount of motion in your video. Ultimately, though, the best way to judge which settings you want is to compare different versions yourself.

ENCODING YOUR VIDEO

Few video-editing packages include Flash Video in their export options – Adobe Premiere Elements and Premiere Pro are exceptions. However, if you're using other editing software, you can encode your finished projects using freely available software. We experimented with various freeware, shareware and commercial FLV encoders and found Adobe Premiere Pro/Elements came top for versatility and the quality of the results. Elements is excellent value at around £50, and Flash Video encoding in both Sorenson Spark and On2 VP6 formats is just one of its many strengths. Its settings are clearly laid out, with precise control over all the variables discussed here.

Allotk Video-to-FLV Converter (www.allotksoft.com) is a shareware program that costs \$40 (around £20) to register. The unregistered version supports videos up to five minutes long. It uses the On2 VP6 codec and includes a range of standard and widescreen resolutions and quality settings, although it's not possible to enter a custom resolution or video bandwidth. We found that selecting the 240x180 resolution, 25fps frame rate and setting Video Quality to Low gave decent results at 250Kbit/s bandwidth.

Riva FLV Encoder (www.rivavx.com) is a freeware program, but its use of the Sorenson Spark video codec means quality isn't quite as good as the encoders mentioned above. It

also falls short of the quality of YouTube's own Sorenson Spark encoder. However, if you want to maintain compatibility with Flash Player 7 it's a decent choice. We found its Bitrate option behaved erratically, so check the size of your files to make sure it's doing what you asked of it; 300Kbit/s equates to 2.2MB per minute. If you decide against the Riva FLV Encoder, it's still worth visiting www.rivavx.com to get hold of the FLV Player, as it's one of the few Windows applications that can play raw FLV files.

EMBEDDING IN AN SWF FILE

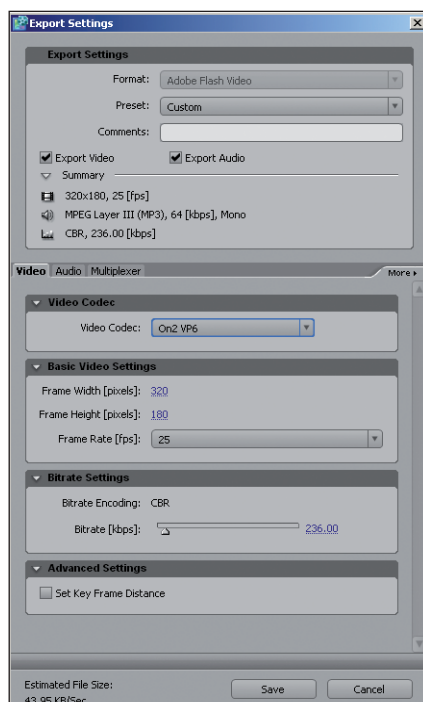
Web browsers don't recognise FLV files, so you'll need to wrap it in an SWF file. Unless you're an experienced Flash programmer, we recommend using the FLV Player at www.jeroenwijering.com. Download the Zip file and extract its contents. It contains various files and folders, including:

- Video.flv** An example FLV file that you can replace with your own
- Flvplayer.swf** The SWF file that contains the transport controls and is inserted into the webpage. This is a locked file that doesn't need to be edited
- Flvplayer.html** An example webpage containing the SWF file in two guises, one with and one without a playlist. The tags it contains define the appearance and behaviour of the player and which FLV file or playlist is used
- Playlist.xml** An XML file that defines which FLV files appear in the playlist
- Preview.jpg** A JPEG image that is shown before playback commences
- Swfobject.js** A JavaScript file that gets rid of the 'Click here to activate' message that plagues Flash movies

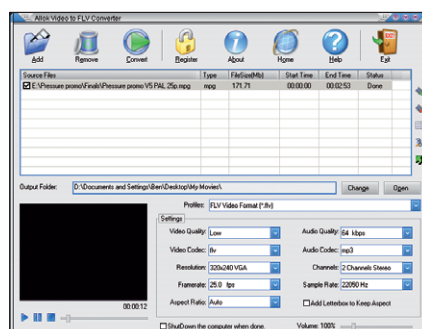
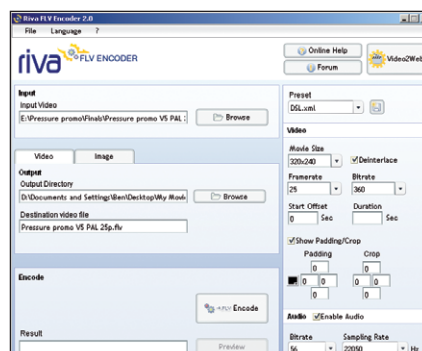
Most settings are made in the Flvplayer.html file. Open it in your HTML editor or Notepad and you'll see the two instances of the player. The top one links directly to Video.flv, while the bottom one links to Playlist.xml. Delete the instance you don't want. There is a vast range of variable parameters, which are well documented in the accompanying ReadMe file. The most important is the name of the FLV file, which you should change to the one you created (if you're using a playlist, add the names of your FLVs in the Playlist.xml file).

Pay attention to the player size as well, which should be your desired width and height for playback, adding 20 to the height for the transport bar at the bottom. Feel free to display your video larger than the encoding size, but be careful to maintain the correct aspect ratio. For example, 16:9 widescreen video with a 400-pixel width will need a height of 245 (400 divided by 16, multiplied by 9, plus 20). If you need to stretch widescreen footage back to the right shape, you'll need to add the following line to the code: `s1.addVariable("overstretch","fit")`. See the ReadMe file for details on this and many other variables, such as colour and transport size, or to show or hide specific controls.

When the FLV player is behaving exactly how you want it, copy the code into your webpage, being careful to maintain the folder structure of the files it links to, and also to copy the JavaScript link in the HTML header. **ES**



▲ Various encoders are available for creating Flash Video files, but the one built into Premiere Elements is the most comprehensive and gave us the best results



▲ Each encoder used a 300Kbit/s bandwidth to deliver the results shown